

Migration from W3100A to W5300

This application note describes what designers and programmers should consider when migrating from W3100A to W5300. For additional information about the W5300, please refer to the W5300 datasheet.

This document contains the following topics:

Section 1, "Comparison"

Section 2, "Hardware Considerations"

Section 3, "Software Considerations"

1. Comparison

1.1 Advantages and Challenges of Migration

This section highlights the various advantages and challenges involved when migrating from W3100A to the W5300. The W5300 provides a higher level of performance while maintaining many characteristics of the W3100A's architecture. Following is a list of the advantages of migrating to the W5300.

- Cost effective.
- Provides a higher level of integration.
- Added Tx free size register and Rx received size register. Users can directly read them and do not need to calculate the value any more.
- TCP sequence and ACK number is automatically processed. Users do not need to calculate the values any more.
- New functions (PPPoE/IGMP/SPI/Keep-alive, etc).
- Supports 8 H/W sockets.
- Supports 16-bit width data bus.
- Flexible Tx/Rx memory allocation.

Following is a list of differences that may present challenges in migrating from the W3100A to the W5300:

- Different package and pin assignment. See Section 2, "Hardware Considerations" for information on addressing these differences.
- Library migration. There is a huge difference between the driver library of W3100A and W5300. See section 3, "Software considerations" for information on addressing these differences.

1.2 Summary and Feature Comparison tables

Table 1 includes information comparing some of the features of W3100A and W5300.

Table 1. W3100A and W5300 Comparison

		W3100A	W5300
H/W	Core	H/W IPv4	
	Voltage	3.3v	
	H/W Reset	High active	Low active
	Ethernet core Clock	25Mhz	
	MCU bus I/F Clock	25MHz or>25MHz	25MHz
	Performance (full-duplex)	20Mbps	80Mbps
	Package	64-pin LQFP	100-pin LQFP
	MCU bus I/F	Direct / Indirect / I2C	Direct / Indirect
	clocked mode for MCU interface	clocked/external/non-clocked mode	Integrated clocked mode
	Tx/Rx memory	16KB	128KB
	Socket number	4	8
	MII	Serial and Nibble	Nibble
Function	TCP/IP protocol	TCP.UDP.IP.ARP.ICMP.MAC	TCP.UDP.IP.ARP.ICMP.MAC. IGMP(UDP Multicasting) PPPOE(ADSL)
	SEND_OK interrupt	Support	
	No delayed ACK	Support in TCP mode	
	Interrupt register clear	By writing "1"	
	shadow register	Support	Removed
	ACK pointer register	Support	Removed
	RD/WR Pointer register	4 byte registers	
	Receive Data Size / Free Tx Buffer Size register	None	Added
	Keep alive command	None	Added
	Direct send command without ARP	None	Added (SEND_MAC command)

2. H/W consideration

2.1 Voltage

Both models are 3.3v device.

2.2 TCP/IP Core

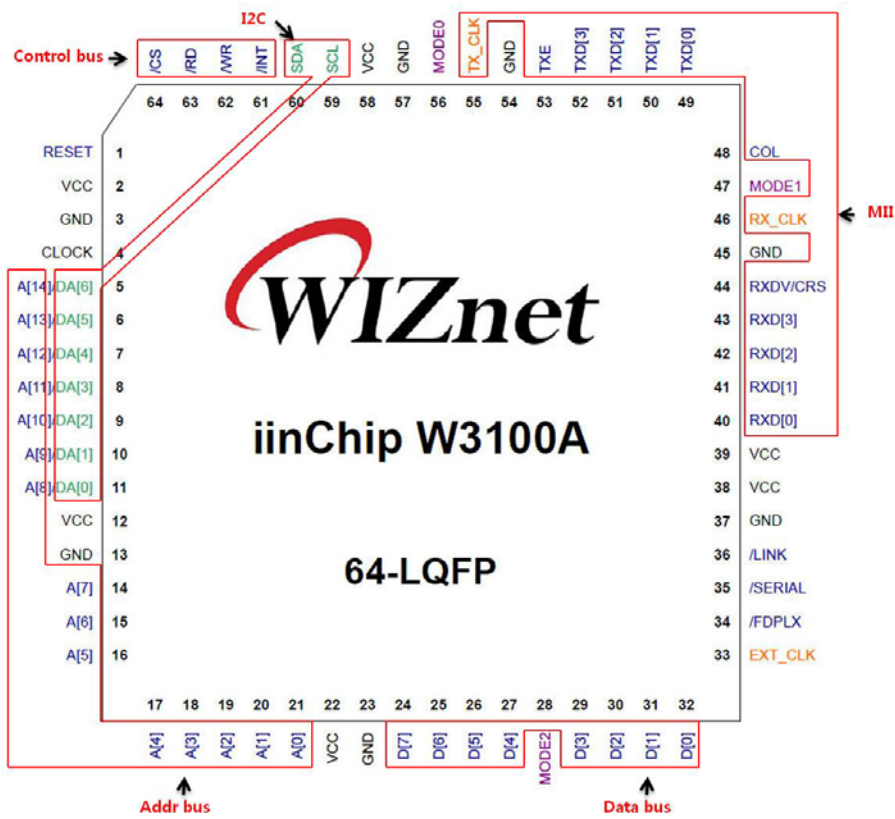
Both models use WIZnet's unique H/W IPv4 core. However, W5300 upgraded this core and fixed weak points which are listed in documentation, "W3100A errata and limitation". Users can open this file by clicking the link below, [http://www.wiznet.co.kr/UpLoad_Files/ReferenceFiles/W3100A_errata_limitation_list1\[2\].pdf](http://www.wiznet.co.kr/UpLoad_Files/ReferenceFiles/W3100A_errata_limitation_list1[2].pdf)

2.3 Package and Pin assignment

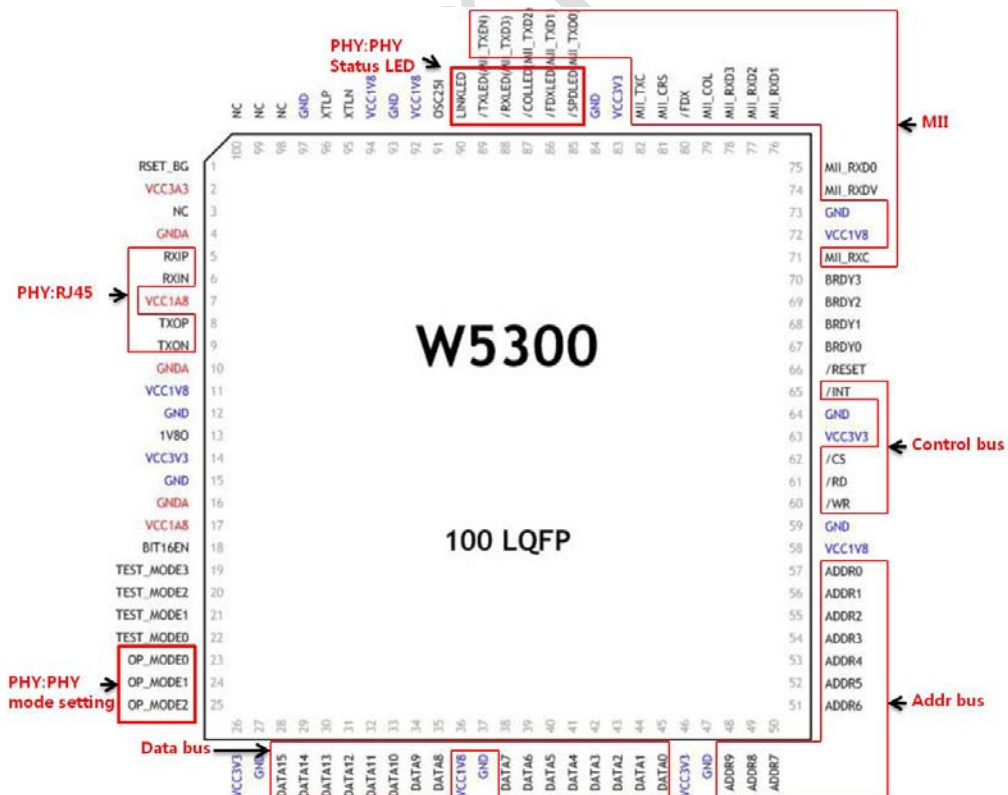
W3100A is 64-LQFP. W5300 is 100-LQFP. Regardless the package or pin assignment, difference is huge. Three main factors made this difference.

- I2C function is removed in W5300
- The width of Data bus is changed to 16-bit in W5300
- PHY is embedded in W5300

There are two parts that users should note when migrate W3100A to W5300. They are MCU interface and PHY interface. MCU interface consists of Control bus, Address bus and Data bus. PHY interface consists of PHY mode setting interface, MII (W3100A/W5300) or RJ45 (W5300) and PHY Status LED. As long as these parts were fully understood, users could easily do the H/W migration from W3100A to W5300.



(a)



(b)

Fig. 1 Pinout considerations when migrate W3111A PLUS to W5300

2.4 Interface

2.4.1 Host interface

In terms of host MCU interface (I/F), W3100A supports Direct, Indirect and I2C, whereas W5300 removed I2C function. Thus, if you were using I2C to communicate with your host MCU, you definitely have to do a new design. If you were using Direct or Indirect, the width of data bus should be set as 8-bit (W5300 supports 8-bit and 16-bit width).

In addition, W3100A provides users 3 modes when use Direct or Indirect way. They are clocked mode, external mode and non-clocked mode. The three options always make users confused and look too complicated, so to make it easy to use, in W5300 chip developers only designed a single reading/writing operation. For more information, refer to timing related chapters in W3100A and W5300 datasheet.

2.4.2 MII interface

W3100A supports two kinds of MII. One is serial MII and the other is nibble MII. As for W5300, it not only has internal PHY, but also supports external PHY. Thus, W5300 must contain MII which is a nibble MII. Thus, if you were using serial MII mode, you have to adjust your PHY chip to the nibble type.

In addition, W5300 has two independent pins: RXDV and CRS. Users should connect pin RXDV of W5300 with pin RXDV of PHY chip, and pin CRS of W5300 to pin CRS of PHY chip.

2.5 System features

2.5.1 Clock

Users can select one of the two clock sources of the W3100A. But W5300 removed the external clocked mode. Thus, W5300 has only one clock source: 25MHz (XTLP/XTLN).

2.5.2 System Reset

Both W3100A and W5300 support H/W and S/W reset.

- For H/W reset signal, W3100A is HIGH active, but W5300 is LOW active.
- For S/W reset, both of them implement this function by writing '1' to the 7th bit in certain registers. W3100A is in CO_CR register (offset 0x00). W5300 is in MR register (offset: 0x00).

2.5.3 Register write/read timing

W3100A supports six kinds of bus I/F mode (three Direct modes + three Indirect modes), whereas W5300 only supports one Direct mode and one Indirect mode. In this case, write/read timing has a huge difference.

When migrating from W3100A to W5300, users should carefully follow W5300's timing. To quick start, users can download W5300 driver code from our website and then study how to implement that timing by writing simple code.

2.5.4 Tx/Rx memory

W3100A has 16KB Tx/Rx memory. W5300 has 128KB Tx/Rx memory. Thus, do not consider the memory limitation when migrating from W3100A to W5300.

2.6 Registers

2.6.1 Register map

In terms of registers, W3100A consists of Control register, Pointer register, System register and Channel register, whereas W5300 consists of Mode register, Common register and Socket register.

Fig.3 shows the Register map of W3100A and W5300.

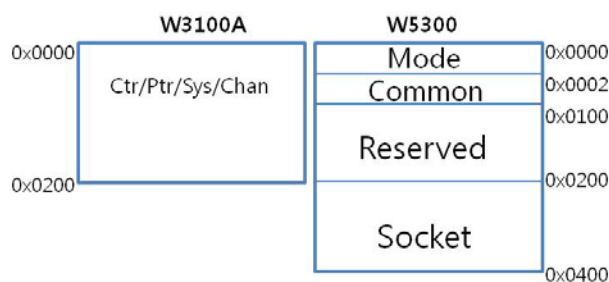


Fig.3 The register map of W3100A and W5300

2.6.2 Register comparison

As a high performance chip, compare to W3100A, W5300 added a few functions, such as Keepalive function, PPPoE function and UDP multicasting function. But more importantly, W5300 optimized many processing methods. For example, 16-bit width data bus is already available and, the allocation method of Tx/Rx memory is more flexible and, the accessing method to Tx/Rx memory is also optimized and, FMTU register may tell you the MTU value of the opposite end when it has a different MTU with W5300 and so on.

In this case, a few new registers are designed and, for other registers some of them maintain the same name but with different functions, at least not fully compatible with W3100A.

Table 2 shows us more about the registers comparison.

Table 2 Register comparison

Type	W3100A	W5300	Comments
Fully compatible (Registers with the same name and same functions)	GAR		
	SHAR		
	SIPR		
	RCR		
	RMSR		
	TMSR		
	IDM_AR		
	IDM_DR		
Partially compatible (Registers with the same name but partially compatible)	IR		
	IMR		
	SMR	SUBR	

functions or with the similar name but partially compatible functions or with the similar name and same functions)	IRTR	RTR	
	Cn_CR	Sn_CR	
	Cn_ISR	Sn_IR	
	IDM_OR	MR	
	Cn_SSR	Sn_SSR	
	Cn_SOPR	Sn_MR	
	Cn_DIR	Sn_DIPR	
	Cn_DPR	Sn_DPORT	
	Cn_SPR	Sn_PORT	
	Cn_IPR	Sn_PROTO	
	Cn_TOSR	Sn_TOS	
	Cn_MSSR	Sn_MSR	
	New designed		MTYPER
		PATR	PPPoE related registers
		PTIMER	
		PMAGICR	
		PSIDR	
		PDHAR	
		UIPR	Unreached IP
		UPORTR	Unreached port
		FMTUR	To get peer's MTU value
		Pn_BRDYR	It configures the PIN "BRDYn" which is monitoring TX/RX memory status of the specified SOCKET.
		Pn_BDPTHR	It configures buffer depth of PIN "BRDYn"
		IDR	Chip ID register
		Sn_IMR	Socket interrupt mask register
		Sn_KPALVTR	Keepalive timer register
		Sn_TTLR	TCP TTL register
		Sn_TX_FSR	Socket tx freesize register
		Sn_RX_RSR	Socket rx received size register
	Sn_FRAGR	Socket fragment register	
	Sn_TX_WRSR	Used to Access to TX/RX memory	
	Sn_TX_FIFOR		
	Sn_RX_FIFOR		

3. S/W consideration

3.1 Software library

Essentially, S/W migration refers to driver migration, which is downloadable through WIZnet website. Thus, for each WIZnet chip, S/W migration is very simple:

3.1.1 How to migrate

<Step1>: Remove W3100A driver

<Step2>: Download W5300 driver

<Step3>: Porting W5300 driver to your host MCU

<Step4>: List up all W3100A's API that your application is using

<Step5>: List up all W5300's API which has the same functions with W3100A's API (See Table 4)

<Step6>: Replace W3100A's API with W5300's API one by one.

<Step7>: Completed

Table.3 Primary APIs comparison between W3100A and W5300

Functions	W3100A	W5300
Initialization of Network	InitW3100A()	iinchip_init();
	setMACAddr()	setSHAR()
	setIP()	setSIPR()
	setgateway()	setGAR()
	setsubmask()	setSUBR()
	getsubmask()	getSUBR()
	GetIPAddress()	getSIPR
	GetGWAddress()	getGAR()
Open Sockets	socket() Note: The symbol of 2nd parameter of socket() is different. See below:	socket()
	SOCK_STREAM	Sn_MR_TCP
	SOCK_CLOSEDM	Sn_MR_CLOSE
	SOCK_DGRAM	Sn_MR_UDP
	SOCK_IPL_RAW	Sn_MR_IPRAW
	SOCK_MACL_RAW	Sn_MR_MACRAW
Basic socket APIs	Listen()	Listen()
	Connect()	Connect()
	Close()	Close()
Socket status	select(i,SEL_Control)	getSn_SSR (i)

Socket Tx free size	Select(i,SEL_SEND)	getSn_TX_FSR(i)
Socket Rx received data size	Select(i,SEL_RECV)	getSn_RX_RSR(i)
Data transmission	send()/sendto()	send()/sendto()
Data receive	recv()/recvfrom()	recv()/recvfrom()
Disable external INT	EX0=0 Note: W3100A driver is 8051 core based/ W5300 driver is Atmel128 based	IINCHIP_ISR_DISABLE() Note: This part depends on your host MCU
Check IR	INT_REG	IICHIP_READ(IR)
Check socket IR	INT_REG&0x01 (check socket 0)	getSn_IR(0-7)
	INT_REG&0x02 (check socket 1)	
	INT_REG&0x04 (check socket 2)	
	INT_REG&0x08 (check socket 3)	
Check socket INT name	I_STATUS[i]&SESTABLISHED	getSn_IR(i)&Sn_IR_CON
	I_STATUS[i]&SCLOSED	getSn_IR(i)&Sn_IR_DISCON
	I_STATUS[i]&SSEND_OK	getSn_IR(i)& Sn_IR_SEND_OK
	I_STATUS[i]&STIMEOUT	getSn_IR(i)& Sn_IR_TIMEOUT
	INT_REG&0x10	getSn_IR(0)&Sn_IR_RECV
	INT_REG&0x20	getSn_IR(1)&Sn_IR_RECV
	INT_REG&0x40	getSn_IR(2)&Sn_IR_RECV
Clear socket IR flag	Direct value assignment Ex: clear SEND_OK flag INT_STATUS(0)=0x20	By setSn_IR() Ex: clear "Connected" flag setSn_IR(s,Sn_IR_CON);
	Clear IR flag	Direct value assignment Ex: INT_REG=0xFF
Enable external INT	EX0=1 Note: W3100A driver is 8051 core based/ W5300 driver is Atmel128 based.	IINCHIP_ISR_ENABLE() Note: This part depends on your host MCU.

3.1.2 Where to download library

W3100A library, errata and reference schematic can be found through below link:

<http://www.wiznet.co.kr/W3100A-LF/download>

W3100A library, errata and reference schematic can be found through below link:

<http://www.wiznet.co.kr/W5300/download>

Preliminary